

Ralf Spenneberg

Linux-Firewalls mit iptables & Co.

Sicherheit mit Kernel 2.4 und 2.6
für Linux-Server und -Netzwerke



 ADDISON-WESLEY

An imprint of Pearson Education

München • Boston • San Francisco • Harlow, England
Don Mills, Ontario • Sydney • Mexico City
Madrid • Amsterdam



27 Nfnetlink und Kernel 2.6.14

Mit dem Kernel 2.6.14 wurde eine neue Architektur in dem Kernel eingeführt: `nfnetlink`. Obwohl der Anwender davon zunächst nicht viel mitbekommen wird, hat sich unter der Haube damit sehr viel getan. Die gesamte Kommunikation zwischen dem Userspace und dem Kernelspace wurde standardisiert und wird von der `nfnetlink`-Schicht übernommen. Dies wurde nötig, da die aktuellen Lösungen drei Probleme aufwiesen:

1. Code-Replikation: `iptables`, `ip6tables`, `arptables` und `eatables` enthalten große Teile identischen Codes, der so nur schwer wartbar ist. Einzelne Code-Teile sind sehr eng mit dem IPv4-Protokoll verkuppelt, wie zum Beispiel das Connection Tracking.
2. Dynamische Regeln: Die aktuelle Struktur unterstützt nur umständlich dynamische Regelsätze, wie sie von einem Intrusion-Prevention-System benötigt werden würden.
3. API für eine GUI: Es existiert keine einheitliche Schnittstelle, über die eine grafische Oberfläche die Funktionen auslesen und Regeln setzen kann.

Um diese Probleme zu lösen, wurde die `nfnetlink`-Struktur geschaffen, und mit `nfnetlink_log`, `nfnetlink_conntrack` und `nfnetlink_queue` wurden IPv4-unabhängige Strukturen geschaffen, die nun auch von anderen Protokollen wie IPv6 genutzt werden können. Dies ist der erste Schritt auf dem Weg zu `pkttables`. `Pkttables` ist ein Langzeitziel des Netfilter-Teams. Mit `Pkttables` soll eine einheitlicher Oberfläche für alle Firewall-Regeln unabhängig vom Layer-3-Protokoll geschaffen werden.

27.1 Der `conntrack`-Befehl

Die wesentliche Neuerung für den Endanwender beim Einsatz der `nfnetlink`-Architektur ist der Befehl `conntrack`, der nun zur Verfügung steht. Mit diesem Befehl können Sie den Inhalt der Verbindungstabelle anzeigen, nach einzelnen Einträgen suchen, neue Einträge hinzufügen und Einträge löschen. Genauso können Sie auch die Tabelle der erwarteten Verbindungen (Expectations) anzeigen und auch hier neue Verbindungen hinzufügen oder löschen.



Achtung

Der Befehl `contrack` ist noch sehr jung und im Moment (Oktober 2005) noch stark in der Entwicklung. Verwenden Sie immer die neuesten Versionen der Bibliotheken und des Befehls von dem Subversion-Server, wenn Sie diesen Befehl testen möchten. Aktuell ist zum Beispiel das Löschen von Einträgen nicht möglich.

Der Befehl `contrack` kann von der Iptables-Homepage heruntergeladen werden (<http://www.Iptables.org/projects/contrack/downloads.html>). Anschließend müssen Sie das Paket nur auspacken und übersetzen. Achten Sie darauf, dass Sie mindestens einen Linux-Kernel 2.6.14 verwenden und vorher die `libnfnetlink`-Bibliotheken installieren.

Der Befehl kann dann sowohl die normale Verbindungstabelle als auch die Expectations-Tabelle betrachten. Sie können die folgenden Befehle ausführen:

- `-L, --dump`: Anzeige der Tabelle.
- `-G, --get`: Sucht nach einem Eintrag in der Tabelle.
- `-D, --delete`: Löscht einen Eintrag aus der Tabelle.
- `-I, --insert`: Fügt einen Eintrag in die Tabelle ein.
- `-E, --event`: Zeigt ein Echtzeit-Protokoll an.
- `-F, --flush`: Löscht die gesamte Tabelle.

Viele der Befehle benötigen zusätzliche Optionen, um die Verbindung zu spezifizieren. Hierfür haben Sie die folgenden Möglichkeiten:

- `-z, --zero`: In Kombination mit dem Kommando `-L` löscht diese Option die Zähler nach der Anzeige.
- `-e, --event-mask <mask>`. Diese Option ist nur gültig in Kombination mit dem Kommando `-E`. Hiermit können Sie die Ereignisse einstellen, die in dem Echtzeitprotokoll angezeigt werden sollen. Die folgenden Events können Sie verwenden: `ALL|NEW|RELATED|DESTROY|REFRESH|STATUS|PROTOINFO|HELPER|HELPIFNO|NATINFO`.
- `-g, --group-mask <mask>`. Auch diese Option ist nur gültig mit dem Kommando `-E`. Sie erlaubt Ihnen die Auswahl des Protokolls (`ALL|TCP|UDP|ICMP`).
- `-s, --orig-src <ip>`: Mit dieser Option geben Sie Quell-IP-Adresse der Verbindung an.
- `-d, --orig-dst <ip>`: Mit dieser Option geben Sie die Ziel-IP-Adresse der Verbindung an.
- `-r, --reply-src <ip>`: Mit dieser Option geben Sie bei einer genatteten Verbindung die DNAT-IP-Adresse an. Ist die Verbindung nicht genattet, ist sie identisch mit der `orig-dst`-Adresse.

- `-q, --reply-dst <ip>`: Mit dieser Option geben Sie bei einer genatteten Verbindung die SNAT-IP-Adresse an. Ist die Verbindung nicht genattet, ist sie mit der `orig-src` Adresse identisch.
- `-p, --proto <protokoll>`: Hiermit wählen Sie das Protokoll (TCP, UDP ...).
- `-t, --timeout <sekunden>`: Hiermit geben Sie den Timeout der Verbindung an.
- `-u, --status <status>`: Hiermit wählen Sie den Status der Verbindung (`[EXPECTED|ASSURED|SEEN_REPLY|CONFIRMED|SNAT|DNAT|SEQ_ADJUST|UNSET]`).
- `-i, --id <id>`: Jede Verbindung hat eine Contrack-ID, über die Sie auf die Verbindung zugreifen können. Die aktuelle Version zeigt jedoch die ID leider nicht an.
- `--tuple-src <ip>`: Hiermit geben Sie bei einer Expectation das Quell-Tuple an.
- `--tuple-dst <ip>`: Hiermit geben Sie bei einer Expectation das Ziel-Tuple an.
- `--mask-src <ip>`: Hiermit können Sie bei einer Expectation die Quell-Maske angeben.
- `--mask-dst <ip>`: Hiermit können Sie bei einer Expectation die Ziel-Maske angeben.

